

CONTROL UNIT FOR A SYSTEM AND A METHOD OF OPERATING A CONTROL UNIT

Field Of The Invention

The present invention relates to a controller for a system and a method of operating the controller.

5

Background Information

A diagnostic device for an engine control unit is known from International Published Patent Application No. WO 97/13064, which describes various diagnostic function modules that are activated by a scheduler. The diagnostic function modules check the systems required for the operation of the engine. The scheduler can block certain diagnostic function modules when the diagnostic function module can be expected to indicate a fault on the basis of faults already discovered.

Summary Of The Invention

The control unit according to the present invention for a system and the method according to the present invention for operating a control unit have the advantage over the related art that the cross-dependencies between running modules are taken into account in activating additional modules. Since the information on the cross-dependencies is provided in a first storage device, this information is compiled centrally. Therefore, additional modules can easily be added or removed again. The mutual dependencies among the modules can be taken into account especially easily for all modules owing to the central storage. This is especially important when calling up a certain module which observes system states that are interfered with by other modules. In such a case, faulty information can be generated due to the observation of interfered with states.

20

It is especially easy to prevent activation of the additional module or to deactivate a module that has already been activated. In the first storage device, only the cross-dependencies between the modules or, in addition, information about disturbances in states of the system and/or observation of states of the system can be stored by

2M360 38789445

the above-mentioned modules. The modules and the scheduler can be implemented especially easily as a software program for a microprocessor. Tables or matrices in the memory of the control unit are expediently used as storage devices.

5 Brief Description Of The Drawings

Figure 1 shows a control unit and the system according to the present invention.

Figure 2 shows an arrangement of the information in a first storage device according to the present invention.

10 Figure 3 shows another arrangement of the information in the first storage device according to the present invention.

15 Figure 4 shows a flow chart corresponding to a method according to the present invention.

20 Detailed Description

Figure 1 shows a control unit 1 which is connected by connecting lines 4 to a system 2 to be controlled. Control unit 1 has a microprocessor 3 which is provided for processing modules 10, 11, 12. Of the modules shown here, module 10 is a function module, module 11 is a diagnostic module, and program module 12 is designed as the scheduler.

25 System 2 may be any technical system that is controlled by a control unit 1. For example, system 2 may be a motor vehicle, an internal combustion engine, or a transmission. Such a system 2 is equipped with a plurality of sensors and a plurality of actuators. Control information generated by control unit 1 can be transmitted over connecting lines 4 to the actuators of system 2 (arrow pointing to system 2). In addition, values measured by sensors of system 2 can be transmitted over the connecting lines to control unit 1 (arrow pointing to control unit 1).

30 Control unit 1 thus receives information about states of system 2, which is processed by control unit 1 and converted to control information for the system 2

in accordance with the desired states (for example, the gas pedal position in the vehicle).

Control unit 1 is represented here only schematically by multiple modules 10, 11, 12 and microprocessor 3. Other hardware components with which those skilled in the art are familiar are not shown. Modules 10, 11, 12 are usually designed as program modules, but hardware components that perform certain functions can also be used. A program module is a sequence of program commands which can be activated or deactivated as a unit by a scheduler. Functions which appear as a unit to the user or are used to control a uniform function can be divided into multiple modules administered separately by the scheduler 12. In addition, only a single function module 10, a single diagnostic module 11 and a single scheduler 12 are shown. A plurality of corresponding modules 10, 11 are of course provided to perform complex control and diagnostic functions. The function modules 10 are necessary for processing the immediate control functions, i.e., they analyze sensor information of system 2 and generate corresponding control information for system 2 according to predetermined setpoints. With a motor vehicle, for example, the input signal of such a function module 10 may be the rpm, and the setpoint information may be the gas pedal position, for which corresponding control information for the engine is then generated. Diagnostic modules 11 have the function of monitoring the proper functioning of system 2. Because of statutory regulations, for example, the functioning of the catalyst in a motor vehicle must be checked during ongoing operation. This is done by activating from time to time a diagnostic module 11 which checks the function of the catalyst by analyzing the signals of the lambda probe.

Function modules 10 and diagnostic modules 11 may be in an active state or an inactive state. In the inactive state, the functions associated with the modules 10, 11, such as diagnosis or control of parts of the system, are not carried out. This means in particular that inactive modules 10, 11 do not receive any data from the system 2 and do not output any information to system 2. In the inactive state, however, the modules 10, 11 can still exchange information with the

microprocessor 3; in particular, it must be possible to exchange information concerning activation of a module (e.g., the possibility of or a request for activation) between the modules 10, 11 and the scheduler 12. Not all modules 10, 11 are needed in all operating states of system 2, so at least some of these modules 10, 11 may be in an active or an inactive state. Individual function modules 10 or diagnostic modules 11 are activated by a sequence control that decides which function modules 10 or diagnostic modules 11 are to be executed. Scheduler 12 is part of this sequence control. The sequence control can make the activation of individual function modules 10 or diagnostic modules 11 dependent upon certain external conditions. For example, function modules 10, which are associated with the ignition or injection of a gasoline engine, are activated as a function of the crankshaft position of the engine. Likewise, diagnostic modules 11 are activated as a function of operating states of system 2. Multiple function modules 10 and multiple diagnostic modules 11 can be processed in parallel, in the sense that the processing of one module is not yet concluded while another module is also being processed at the same time. One problem with this is that certain modules supply faulty information when certain other modules 10, 11 are activated at the same time. For example, activated carbon filters are provided with modern vehicles to prevent vaporized hydrocarbons from the tank from entering the environment. Then for a limited period of time in ongoing operation of the engine, air is taken in through this activated carbon filter, and the hydrocarbons present in the filter are burned in the engine, so that the filters are cleaned again by this measure. However, this is associated with interference in the lambda probe signal, so that diagnosis of the catalyst is impossible during the cleaning of the activated carbon filter. Thus, diagnostic module 11 for checking the catalyst must not be activated while function module 10 for cleaning the activated carbon filter is activated. In addition, function modules 10, e.g., for carburation, are also interfered with during this cleaning.

To take into account this cross-dependency between the modules 10, 11, the present invention calls for a scheduler 12 which takes these cross-dependencies into account by analyzing storage device. Figure 2 shows the content of a first memory

to which the scheduler has access. X, Y and Z in Figure 2 denote physical states of the system 2 which have an influence on sensor signals of the system 2. Column I lists three different states of system X, Y and Z. These system states are either influenced or observed by program modules, labeled here as A, B, C and D.

5 Furthermore, a module may observe a state and influence it at the same time (module D appears in column II and III). When a module influences a state in such a way that other modules observing this state supply faulty results because of this influence, this is interference between these modules which comes about indirectly due to interference in a state by at least one of the modules. For example, when a 10 state is needed as a necessary parameter for a calculation by a module, a misdiagnosis or faulty control pulses may occur due to interference with other modules.

15 Column II lists the modules which observe the corresponding states, and column III lists the interfering modules. State X, for example, is observed by program modules A and B and is interfered with by program module C. State Y is observed by module A and interfered with by module B. State Z, for example, is observed only by module A and is not interfered with by any module. When module A is to be activated, scheduler 12 accesses the tables shown in Figure 2 and ascertains that 20 module A observes state X, state Y, and state Z. Since states X and Y are interfered with by modules C and B, module A is enabled for activation only if modules C and B are not activated. In addition, module A also observes state Z, which is not, however, influenced by any other module. This could be useful for 25 expansion of the control unit; in particular, other modules which might have an influence on state Z can easily be entered in the table created in Figure 2.

30 The information stored in Figure 2 can be stored in any storage device or in any other form, e.g., as a matrix. It is essential only that a storage device be provided on the basis of which it is possible to determine which modules have cross-dependencies, i.e., which modules must not be activated simultaneously on the basis of existing cross-dependencies. It is also important that function modules 10 or diagnostic modules 11 can be added to or removed from the control unit 1 easily

by deleting or adding the individual modules and making a suitable entry in the storage device. In addition, complicated cross-dependencies between different modules can thus be taken into account in a very clear way.

5 The table shown in Figure 2 contains an excess of information, or a certain redundancy, inasmuch as only columns II and III are actually needed to determine cross-dependencies between the modules. The reason for column I is that it makes reconstruction of the cross-dependencies especially simple. This simplifies the addition of new modules, because it is usually known which states a new module needs (column II) and which states a new module interferes with (column III).
10 Thus, as an alternative, only columns II and III may be stored in the first storage device. Figure 3 shows a corresponding representation. The representation in Figure 3 shows the same information as columns II and III in Figure 2 in a different representation. Column IV lists the modules, and column V lists the interfering modules. Figure 3 shows that module A is interfered with by modules B, C and D, module B is interfered with by module C, module C is not interfered with by any other module, and module D is interfered with by module B. The table according to Figure 3 may be part of the fixed program, or it may be created in initialization the computer. According to the table in Figure 3, two modules interfere with one another when at least one of the modules is entered in column IV and the other is entered in the same line in column V.

15
20
25 Figure 4 shows schematically an example of the program sequence of scheduler 12 which accesses the information according to Figure 2. In a first program step 101, a request is sent to the scheduler 12 for a certain module, e.g., module A here, to be processed in accordance with a state of system 2 or another condition, such as the fact that a certain period of time has elapsed. A subsequent program step 102 determines which states are observed by module A (states X, Y and Z) by polling the first storage device, as shown in Figure 2. Another subsequent program step 30 103 determines which modules (B, C, D) interfere with these states (X, Y and Z). In addition, step 102 determines by polling the first storage device which states are interfered with by module A (none here). Program step 103 also checks to see

which modules are observing the states so determined. Then by polling additional storage device, a subsequent program step 104 determines which of these modules which are interfered with by A or are in turn interfering with A (modules B, C, D) are currently active, and then on the basis of this information, a subsequent step 5 105 decides whether module A may be activated. If no interfering module (B, C, D) is active, and if no activated module is interfered with by activation of A, then the activation of module A is enabled so that module A is processed. Then another program step 106 notes in the second storage device that module A is activated. Various options are conceivable if a conflict with other modules that have already 10 been activated occurs due to activation of module A. For example, scheduler 12 may then refuse to activate module A in step 105, so that module A would have to try again to report to the scheduler 12 at a later time. Another possibility is for scheduler 12 to decide in step 105 to inactivate an interfering module and then 15 allow activation of module A. This can be decided, for example, on the basis of priorities assigned to the individual modules. In subsequent program step 106, the information regarding which modules are currently activated is then stored.